NASA/CR—2002–211858

CATS-based Agents That Err

Todd J. Callantine
San Jose State University

November 2002

NASA/CR—2002–211858

CATS-based Agents That Err

Todd J. Callantine
San Jose State University

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

November 2002

Acknowledgments

This research was supported by the System-Wide Accident Prevention element of the FAA/NASA Aviation Safety Program.

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
301-621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

Introduction

This report describes intelligent agents that make errors. The agents extend previous efforts to develop agents based on the Crew Activity Tracking System (CATS) (Callantine, 2001; Callantine, 2002b). The eventual aim of this research is to use agents that err in realistic ways to assess safety risks in complex systems.

Researchers have long sought to understand the role of human error in complex system incidents and accidents. In aviation, especially, a variety of error taxonomies and methods for using them to understand errors have been developed; however, it is not always clear how such techniques should be used to prevent future errors (Wiegmann and Shappell, 2001). Researchers who have focussed on accident investigation admit that “waiting until tragedy strikes to employ the taxonomy is certainly not the best tactic to follow” (Shappell and Wiegmann, 1997, p. 289). They propose that error taxonomies can also be useful for identifying features within the work environment that can lead to incidents and accidents in the future. This is no doubt true, but analyzing contextual factors that could create a particular error chain can lead to an ‘explosion’ of ‘what-ifs.’ Thus, safety practitioners need a principled way to assess factors that cause errors and the resulting impact on system safety.

Some researchers have addressed this problem through the development of error analysis techniques that impart structure to the process. For example, Fields, Harrison, and Wright (1997) have developed Techniques for Human Error Assessment (THEA) to identify potential errors from system usage scenarios. This technique is in some respects similar to the Cognitive Walkthrough method of usability analysis (Polson, Lewis, Rieman, and Wharton, 1992), though its scenario-centered approach purportedly avoids difficulties encountered in applying the Cognitive Walkthrough technique (Pocock, Harrison, Wright, and Johnson, 2001). Other researchers are applying formal methods to system and interface design and analysis (e.g., Degani and Heymann, 2000). Such approaches formalize ‘traditional’ human factors techniques—matching information requirements to task demands, while respecting human limitations—for complex dynamic systems instead of simple human-computer interactions.

Another approach is to use models of human performance to simulate plausible human errors, and examine the effect of these errors on the human-machine system as a whole. This new approach first requires validated computational human performance models that represent operators of complex systems at a suitable level of fidelity. Such models must then be enhanced to include mechanisms for creating realistic errors. Computer-based agents that incorporate such models should make errors with a frequency that is low enough to correspond to empirically observed error frequencies, but also high enough to provide substantial effects to analyze. In a mature implementation of this methodology, agents would interact, closed-loop, with a fast-time simulation of the controlled system numerous times, Monte Carlo fashion, and researchers would assess the results of the trials with a focus on how the simulated errors impact system safety. Because the simulated errors are attributable to specific processing mechanisms in the human performance error model, safety practitioners would be aware of the factors involved. Safety practitioners would then implement training enhancements and/or system design modifications based on these results.

This report presents preliminary research on such a methodology that extends research on CATS-based air traffic controller agents (Callantine, 2002b). The remainder of the report is organized as follows. It first describes existing frameworks that have been used to investigate errors. It then discusses the CATS-based air traffic controller agents, and presents modifications that enable the agents to make errors. It compares the performance of the agents to agents that do not purposefully err. The report concludes with a discussion of the viability of the error simulation approach, capturing and assessing the impact of realistic behaviors such as error-detection and remediation, and future research directions.

Error Frameworks

This section provides background on error taxonomies developed to understand and classify human errors. It discusses several frameworks and their theoretical underpinnings. It also discusses how these frameworks have been applied to analyze errors, where applicable.

Research on aviation accident investigation provides a starting point. O'Hare, et al., (1994) and Wiegmann and Shappell (1997) both sought ways to improve analyses of post-accident data. In their search for a useful general error-classification framework, Wiegmann and Shappell (1997) examined three principle frameworks: (1) the Human Information Processing model of Wickens and Flach (1988); (2) Rasmussen's (1982) Model of Internal Human Malfunction, and (3) Reason's (1990) Model of Unsafe Acts. O'Hare, et al. (1994) adapted the Model of Internal Human Malfunction to classify errors, and concluded that extensions to include a variety of contextual factors were desirable. Shappell and Wiegmann (1997) went on to develop the Taxonomy of Unsafe Operations to include such factors, using the Model of Unsafe Acts as their starting point.

This section presents characteristics of these error frameworks, together with several others. These include Norman's (1992) Categorization of Action Slips, the cognitive failure modes used in THEA (Fields, Harrison, and Wright, 1997), the situation awareness-based error taxonomy of Endsley (1999), Funk's (1991) error taxonomy for cockpit task management, and the error phenotypes discussed by Hollnagel (1991). The original references, together with Weigmann and Shappell's (1997) review, provide considerable detail. Therefore, this review omits all but the essential characterizations required to compare and contrast the different schemes.

Human Information Processing model

Wiegmann and Shappell (1997) note that most models of human error derive from human information processing theory, and the model of Wickens and Flach (1988) is a representative case. The model is expressed as series of stages through which an external stimulus is captured, transformed into useful information, and used to support decision making. The eventual decision is then translated into a response and executed. After initial stimulus capture, finite attentional resources moderate each processing stage. Memory supports the decision-making process.

Based upon this model, errors are classified according to the information processing stage in which they are likely to have occurred. By applying this taxonomy to a database of military accidents, Wiegmann and Shappell (1997) show that errors in decision making are more frequently associated with serious accidents. Response execution errors, on the other hand, are more frequently associated with minor accidents.

Model of Internal Human Malfunction

Rasmussen's (1992) Model of Internal Human Malfunction similarly has its origins in information processing theory, but with a slightly different definition of stages from the Wickens and Flack (1988) model. O'Hare et al. (1994) operationalize the Model of Internal Human Malfunction as a step-wise diagnostic algorithm for attributing accident causation as follows. First, if there is no opportunity for the operator to intervene, then the error is due to the system, and not an operator error. Next, if the operator misses a cue, the error is an *information error*. Next, a *diagnostic error* occurs if the operator fails to correctly diagnose the state of the controlled system. The operator may then commit a *goal error* if she chooses an unreasonable goal based on a correctly diagnosed system state. Next, if the operator fails to identify an appropriate strategy for achieving the goal, the error is a *strategy error*. A *procedure error* may then occur if the operator selects a procedure inconsistent with the strategy. Finally, an *action error* results from improper execution of the correct procedure.

O'Hare et al. (1994) use this model to identify serious crashes in a database of incidents and accidents with goal errors and diagnosis errors, while 'minor mishaps' are most closely associated with procedural errors. Wiegmann and Shappell's (1997) analysis of military accidents yields similar results: major accidents are associated with goal and strategy errors, while minor accidents are attributable to procedural and action errors.

Model of Unsafe Acts

Reason's (1990) Model of Unsafe Acts first divides all errors into intended and unintended acts. Two *basic error forms*, called *slips* and *lapses*, are unintended. Errors in intention are *mistakes*. (The model also includes *violations*, which are intentional actions the operator knows to be wrong.) Slips are attentional failures, and include *intrusion, omission, reversal, misordering*, and *mistiming*. Lapses are memory failures, and include *omitting planned items, losing one's place*, and *forgetting intentions*. Both slips and lapses are associated with skill-based behavior. Mistakes, on the other hand, are associated with rule-based and knowledge-based behavior. Rule-based mistakes involve either *incorrect application of a good rule*, or *application of a bad rule*. Knowledge-based mistakes arise from a faulty mental model of the problem space.

Interestingly, in Wiegmann and Shappell's (1997) analysis using this error framework, major accidents were associated with violations, while minor ones were associated with mistakes, slips, and lapses. Shappell and Wiegmann (1997) chose this error framework for extension into their Taxonomy of Unsafe Operations, in which they add layers of *unsafe conditions of the operator* and *unsafe supervision*. These additional contextual factors enable Shappell and Wiegmann (1997) to capture a wide range of causal factors underlying an actual aircraft accident.

Situation Awareness-Based Error Taxonomy

Beyond the purview of Shappell and Wiegmann, Endsley (1999) presents an error taxonomy derived directly from her theory of situation awareness. She identifies three levels of situation awareness: (1) identifying key elements of the situation, (2) comprehending them in light of operational goals, and (3) projecting them into the future. Her error taxonomy begins at the first level of situation awareness with *failure to correctly perceive information*. This level encapsulates system-derived problems, such as *data not available* and *data hard to discriminate or detect*, as well as human malfunctions, including *failure to monitor or observe data, misperception of data*, and *memory loss*. At the second situation awareness level is *failure to correctly integrate or comprehend information*. This includes *lack of*, *poor*, and incorrectly applied mental models, *over-reliance on default values*, and a catch-all category, *other*. At the third situation awareness level, *failure to project future actions or state of the system* also concerns poor or missing mental models, *over-projection current trends*, and *other*. To these levels, Endsley (1999) adds a high-level category, called *general*, which includes *failure to maintain multiple goals* and *habitual schema*. This taxonomy is intended to help specify a situation awareness-centered design methodology.

Categorization of Action Slips

Norman's (1981) action slips cover unintentional errors committed within a cycle of action, while a 'mistake' is an error of intention (Norman, 1988). Norman's action cycle is akin to other information processing models, beginning on the 'evaluation side' with a world state to be perceived. The human then interprets this perception, and evaluates the interpretation to arrive at goal. On the 'execution side,' once the human has formed the goal, she must form an intention to act, specify the required sequence of actions, and execute these actions on the world. The way in which the 'unit of analysis' shifts among steps of the action cycle makes Norman's view of human problem solving one of the most fluid; the description evokes a blur of processing stages in which feedback from a single action 'catches up' with the intentions that drove it.

This characteristic of Norman's stages of action supports the notion of 'activation' that underlies some of his slip categories. The most basic slip is the *loss-of-activation error*, in which the human forgets an intended action. In the *associative activation error*, a related activity interferes with the intended one. A *description error* occurs when the human performs the correct action on the wrong, albeit similar, object. A *capture error* occurs when a more frequently performed action overrides the correct one. When external data in the environment obscures the action to be performed, a *data-driven error* occurs. Finally, a *mode error* can occur when a modal system is involved—the human performs the correct action for one mode when the system is actually in another mode.

Other frequently mentioned error forms, such as *sequence errors*, where actions are performed in the incorrect order, and *post-completion errors*, where the last step of a procedure is forgotten, derive from the basic set identified by Norman. While Norman's slips are assumed to occur on the execution side of the action cycle, some, such as the data-driven error, may also involve faults on the evaluation side of the cycle. The sequence error also involves evaluation faults. As with other taxonomies, these error classifications are equivocal; if sufficient context information is not available, a given error can possibly be interpreted as one of several types of slips using Norman's framework.

Cognitive Failure Modes

Fields, Harrison, and Wright also state that "errors can be regarded as failures in cognitive processing" (1997, p. 16). They identify cognitive failure modes from an abridged action-cycle model derived from Norman's. Specifically, goals may be incorrectly triggered—either the goal is wrong, or the right goal is triggered at the wrong time—or an active goal may be lost. Moreover, goals may not be achievable in the current operational context, or goals may be in conflict. The plan formation stage in the action cycle may generate a faulty or impossible plan. After planning, actions are subject to slips and lapses. On the evaluation side of the action cycle, the human may fail to perceive information correctly, or misinterpret information that is correctly perceived.

Cockpit Task Management Error Taxonomy

Funk (1991), in his theory of cockpit task management, presents yet another error taxonomy based on an 'algorithmic' view of the cockpit task management process. Step one is to *create the initial agenda*. Then, *until the mission goal is achieved or deemed unachievable*, the flight crew performs the following steps: *assess the current situation*, *activate tasks whose initial events have occurred*, *assess status of active tasks*, and *terminate tasks with achieved or unachievable goals*. For active tasks, the flight crew should prioritize them, and allocate resources on the basis of priority: *initiate newly activated high-priority tasks*, *interrupt low-priority tasks if necessary*, and *resume interrupted tasks, when possible*. Finally, the flight crew should *update the agenda* and repeat the cycle.

Errors can occur when initiating, assessing, prioritizing, interrupting, resuming, and terminating a task, or when allocating resources to a task. For example, the flight crew can fail to initiate a task, initiate the wrong task, or they can initiate the correct task early or late. Similarly, the crew can allocate too many or too few resources to a given task. Some of these categories are only viable in retrospect; for example, until the correct task is initiated late, the crew can only be said to have failed to initiate the task.

Erroneous Action Phenotypes

Finally, research by Hollnagel (1991) on phenotypes for erroneous actions focuses less on generative mechanisms, and more on the characteristics of observed errors. Hollnagel identifies five simple error phenotypes: *omission, replacement, intrusion, repetition,* and *reversal*. When the timing of such errors is considered, such errors entail *absence of action* or *unexpected action*. These basic phenotypes lead to a taxonomy of phenotypes of erroneous actions comprised of four main categories: *action in wrong place*, *action at wrong time*, *action of wrong type*, and *action not included in current plan*. Each of these has lower-level phenotypes, such as *jumping forwards*, *delay*, *premature action*, or *insertion*. The application of the taxonomy is governed by several assumptions, including the a priori definition of plans, the inability to retract actions, and the requirement that all the actions in an action sequence must be performed.

Summary

This review of error taxonomies indicates the manner in which errors are classified depends on an underlying model of human behavior in context. The extent to which a model captures the surrounding context to some extent determines the applicability of the taxonomy based on it. As every model may be viewed as having a particular purpose, structure, content, and specificity (Jones and Mitchell, 1987), so may a derivative error taxonomy. Certain taxonomies classify errors based on task-oriented model of interaction of with a complex system. Examples are O'Hare et al.'s (1994) application of the Model of Internal Human Malfunction, and Funk's (1991) cockpit task management error taxonomy. Others classify errors based on a cognitive model of the operator (e.g., Wiegmann and Shappell's (1997) human information processing taxonomy). General 'action cycle' views also suffice (e.g., Norman, 1981). Hollnagel's (1991) error phenotypes are perhaps least bound to such a model, but they too assume a basic structure for the interaction, in terms of some normative, time-anchored sequence of activities. In general, to classify errors, one must first have a model of a correct way for humans to behave. Any errors that result are then due to faults in the process as modeled. Cognitive models enable the attribution of faults to specific processing problems (e.g., cognitive resource limitations); task-oriented models work in conjunction with analysis techniques (e.g., THEA) to provide insights on causation.

A recurring theme in error taxonomies is the distinction between mistakes as errors of intention, and slips or lapses as unintentional errors. This holds even when the distinction is not explicit. For example, slips could conceivably cause most of the errors in the sequence given by the Model of Internal Malfunction (information error, diagnostic error, goal error, strategy error, procedure error, action error). However, there is a strong implication that information and action errors are due to slips, while the rest are at a higher rule- or knowledge-based level associated with mistakes.

CATS ATC Agents

Each of the above error taxonomies is useful for thinking about human error in support of the present research. They provide a basis for classifying observed errors, and for understanding processes by which errors are generated from faults in an underlying model of behavior. This research examines one possible architecture and processing scheme, together with underlying knowledge representations, from which a 'model' of behavior for intelligent agents emerges. Failures in processing this model engender errors. This section describes how agents that function as air traffic controllers are used as a test bed for agents that err. Callantine (2002b) provides a detailed description of the air traffic controller agents used as a starting point for this research (referred to below as the 'nominal agents'). Both the nominal agents and the agents that err are implemented as Java™ application programs. Both are based on a CATS model of air traffic controller activities, with underlying skills and control rules. After an overview of the nominal agents, this section describes the mechanisms by which the agents err.

The nominal agents extend the Crew Activity Tracking System (CATS) modeling framework by using the CATS model to structure the overall air traffic control (ATC) task. When performing a particular ATC activity, the agents access an underlying 'skill library' and a set of 'control rules.' The skill library contains methods that perform perceptual skills, like identifying the aircraft 'in front of' another, or determining the value of the heading vector to issue as a clearance. The control rules generally represent ATC strategy selection, such as whether to issue a speed clearance or a heading clearance in a particular situation. At a lower level, the control rules specify how the selected clearance should be constructed (i.e., which skills to access in order to formulate the clearance). In many cases, rather than directly issuing a clearance, the agents first construct a plan for addressing a particular control problem. An overall plan may consist of several steps. The control rules

- **Maintain situation awareness**
  - **Monitor traffic display**
  - **Scan aircraft**
- **Determine aircraft to work**
- **Manage handoffs**
  - **Accept aircraft**
    - **Accept handoff**
    - **Roger check-in**
  - **Initiate handoff**
    - **Inform other controller**
    - **Issue frequency change**
- **Manage descents**
  - **Issue descent clearance**
- **Manage separation**
  - **Evaluate separation clearance options**
  - **Issue separation clearance**
- **Manage spacing**
  - **Evaluate spacing clearance options**
  - **Issue spacing clearance**
- **Manage nonconformance**
  - **Re-issue clearance**

Figure 1. CATS activity model for ATC task.

also specify conditions under which the agent should execute a particular step of a plan, adapt it, or abandon it altogether. The agents use the high-level CATS model shown in Figure 1. Because the model includes a 'determine aircraft to work' activity that embodies a prioritization scheme for selecting 'control problems' to address, the nominal agents exhibit a characteristic 'flow of control' (Figure 2). This flow can be viewed as a sort of 'action cycle' for the agents. Basically, the cycle begins with determining which aircraft are under the agent's control. Next, the agent assesses the traffic to determine what 'control problems' currently exist. Agents then select, according to the flow of control depicted in Figure 2, the highest priority problem and address it. Addressing a problem may, for example, create a plan whose execution conditions are met on a subsequent cycle, at which time the agent executes the plan (more detailed examples of how the nominal agents work are provided in Callantine, 2002b). When controlling traffic in a reasonably busy ATC sector, an agent may identify numerous control problems. In this situation, the agent will address each one in turn, in the order that corresponds to the flow of control in Figure 2.

Central to the nominal agents are the 'beliefs' each agent maintains. Beliefs are simple representations of the current task context (e.g., 'know which aircraft to clear'), and the traffic situation (e.g., 'conflicts (AAL497 AAL508) (UAL1043 DAL323)' or 'check_within_flow_spacing 1693 (TWA292 UAL649)'). The task context beliefs specify what activity to perform next, while the situation beliefs support determinations about which aircraft or set of aircraft the activity shall address. Beliefs beginning with 'check_' represent control problems that an agent has just addressed and should not address again until the specified time. The agent skips these problems until after the specified time, to allow time for any clearances it issued to

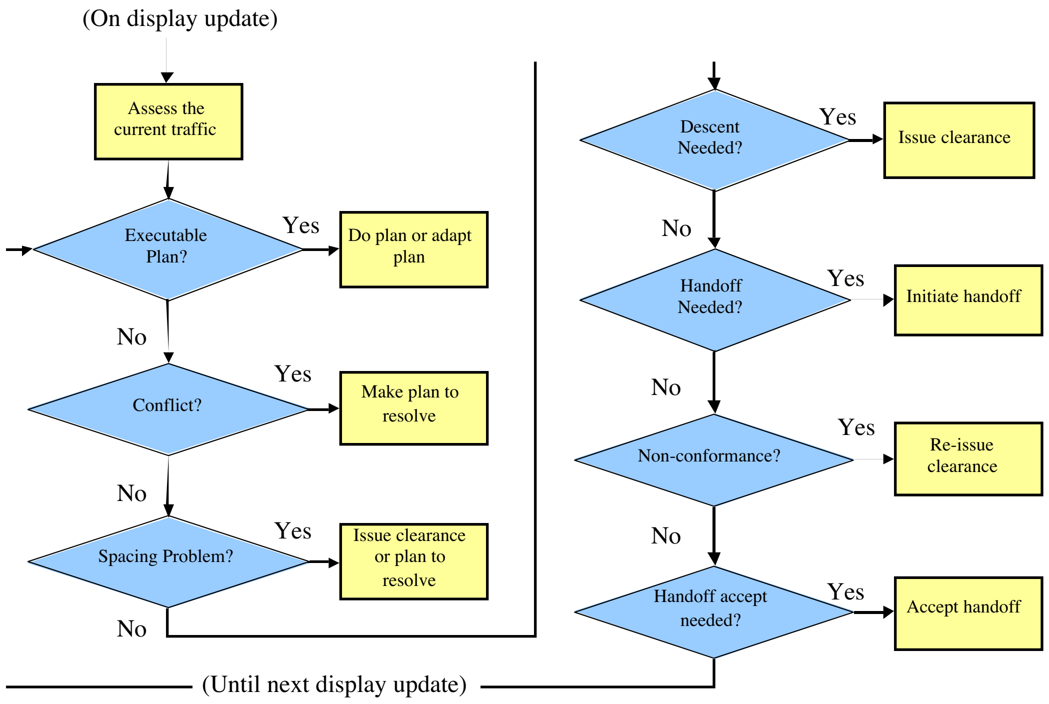

Figure 2. Flow of control for nominal CATS ATC agents.

solve the problem to take effect, and to make time for addressing lower-priority problems. In general, beliefs specify the context that drives the agent's behavior. One may view most beliefs to represent perceived information, and information in working memory, while 'check_' beliefs represent retrospective memory about problems that have been addressed. Agents implement a model of prospective memory via plans.

An agent adds or removes beliefs from its current set as it performs activities. The first sub-activity of 'maintain situation awareness,' 'monitor traffic display,' (see Figure 1) produces a belief that is a list of aircraft currently available for the agent to control. The next sub-activity, 'scan aircraft,' uses the belief about available aircraft to produce a series of beliefs that indicate the set of control problems that the agent needs to address. The next activity, 'determine aircraft work' retrieves a belief about the set of aircraft in the next control problem to address, and so on.

When an activity such as 'evaluate separation clearance options' focuses on a given control problem, a 'check_' belief is created for that set of aircraft, as described above.

When performing a particular activity involving formulating a clearance or planning a clearance, an agent accesses its skill library and control rules. Control rules are specified such that two aircraft are used as parameters: the 'front' aircraft, and the 'back' aircraft (see Callantine, 2002b). Given these two aircraft, rules typically access some information about one or both of them to formulate the clearance or develop the plan. For example, a rule may require information about the current speed or heading of the leading aircraft, or the speed performance range of the following aircraft. When a clearance is formulated and ready to be issued, the agent issues it by executing one of 'issue clearance' activities (i.e., 'issue separation clearance' or 'issue spacing clearance') shown in Figure 1.

This description is considerably abbreviated from that provided in Callantine (2002b), but it covers all the aspects of agent processing in which faults are introduced to produce errors. Four general mechanisms were developed to alter nominal agent function by introducing processing faults. The following subsection identifies these mechanisms, describes how they are implemented to cause the agents to err, and discusses the classes of errors generated in terms of the error taxonomies above.

Error Mechanisms

The present research seeks a process by which the nominal agents can make *realistic* errors. Like the error taxonomies, the process is intimately tied to the agent architecture and processing scheme that serves as a starting point. It is also tied to the domain of application. This research seeks error mechanisms that lead to the sorts of errors air traffic controllers have been observed to make, and that can be explained as faults in the cycle of air traffic control activity. To this end, four general error mechanisms were identified. First, an agent may probabilistically drop ('forget') a belief, or confuse aircraft represented in a belief with other aircraft. Second, an agent may confuse the 'front' and 'back' aircraft when applying a control rule. Third, an agent can 'misread' displayed information about an aircraft, or 'incorrectly recall' information about it. Fourth, an agent can confuse the clearance type and contents when issuing it.

The first error mechanism—probabilistically dropping a belief, or confusing the aircraft represented in a belief with other aircraft—operates each time an agent attempts to add a belief to its belief set. As with all the error mechanisms, a probability value of .05 is used to trigger an error. However, as implemented, this mechanism actually introduces repeated trials to produce an error. First the agent may 'forget' an aircraft with probability .05. Failing that, the agent may confuse a callsign with probability .05. The process of confusing a callsign itself involves a cascade of repeated trials. First, the mechanism attempts to confuse a callsign with one of the same carrier and two shared digits with probability .05. Failing that, the agent may confuse a different carrier with two shared digits with probability .05, and so on. Thus, the error mechanisms are designed with rules that cause the agents to commit the most likely errors first. As a callsign-confusion example, the error mechanism tries first to confuse the aircraft AAL233 with AAL633 before it attempts to confuse it with UAL933. Another valid aircraft that meets the specified criteria must be present for the operation to succeed. Note that dropping a belief can lead to an error of omission, if the dropped belief concerned a high-priority control problem. The proximity of two aircraft on the traffic display is not considered when confusing callsigns.

The second error mechanism—confusing the front and back aircraft in a control rule—simply swaps the two aircraft with probability .05. This mechanism makes it possible for an agent to formulate a clearance or planned clearance for the wrong aircraft, with reference to the wrong aircraft. Thus, for a speed clearance, the two aircraft may close on each other instead of separate; for a heading vector, the two aircraft may turn toward each other instead of away (although the geometry of the conflict determines the actual effect).

The third error mechanism pertains to reference and recall of incorrect details about an aircraft with a control rule. This mechanism was implemented solely with respect to speed clearances. The control rules for such clearances typically require referencing the speed of one aircraft, and recalling the performance range of the aircraft to clear, so that the rule produces a speed clearance the aircraft can fly. Both of these operations can fail with probability .05. For example, consider a case where an agent is applying a rule that implements the strategy of clearing an aircraft to the same speed as the aircraft in front of it to maintain spacing. The agent can 'misread' the leading aircraft's speed from the display, and/or incorrectly recall the valid speed range of the clearance aircraft.

The last error mechanism operates after an agent formulates a clearance. When a clearance value for a heading, speed, or altitude could also be a value for a different clearance, an alternate clearance type is substituted with probability .05 (e.g., instead of 'slow to 240,' 'fly heading 240'). Therefore, this error mechanism includes rules for identifying which clearance values make sense for clearances of different types. The mechanism will not, for example, confuse a heading of 080 with a speed clearance to 80 knots because such a speed clearance value is inordinately low.

Any of these error mechanisms may also affect an agent's plans. An agent can make plans for the wrong aircraft, with the wrong clearance values. The agent may forget that an aircraft has an executable plan. Even if none of these errors occurs, the agent may yet confuse the clearance type when issuing the planned clearance.

Implementing this set of error mechanisms would seem to provide a plethora of problems for the agents. However, the mechanisms were selected in part for their benign effect on overall agent performance: these sorts of errors, while producing incorrect behaviors, do not 'break' the agents (i.e., they do not cause the agents to 'crash'). This effect would clearly be undesirable, as one cannot analyze the effects of errors on the overall system when the system does not continue to operate for a period of time after an error is made. A variety of error-tolerance mechanisms in the baseline agents prevent such fatal errors. One key example is that of the representation of operational constraints on individual aircraft that the agents use (Callantine, 2002a). If a clearance does not make sense according to the rule base used to update the constraint representation, the agents will not issue the clearance. For example, if an erroneous plan entails sending an aircraft direct to a point on its planned route of flight that does not exist (because it is really part of another aircraft's flight plan), the agent drops the plan.

When they succeed in producing errors that have tangible effects, the four basic error mechanisms can 'chain together' to create compound errors. For example, an agent may first to address a control problem that has already been incorrectly specified. In this case, the agent is already working with the wrong aircraft. It can then confuse the 'front' and 'back' aircraft, incorrectly read or recall information about one or both, and finally issue the wrong type of clearance (e.g., a heading clearance, when speed was intended). Because the probability of an error on any given trial is .05, however, the overall probability of this type of effect is reasonably remote.

The error mechanisms enable the agents to make a variety of errors. On their face, they may appear to be simply slips or lapses—indeed, many are. However, these mechanisms may actually produce errors that may be classified as mistakes. For example, the first error mechanism may generate a dropped belief. However, if the belief that was ‘forgotten’ actually concerned the top-priority control problem, the mechanism has effectively generated a ‘goal’ or ‘strategy error,’ in the sense of the Model of Internal Human Malfunction. As another example, planning for the wrong aircraft, or forgetting a plan, maps to ‘faulty or impossible’ plans included in Fields, Harrison, and Wright’s (1997) cognitive failure modes taxonomy.

Because the nominal agents described in Callantine (2002b) require further research and validation, they likely already make mistakes involving incorrect control rules or incorrect rule applications, and incorrect prioritization of control problems. They are somewhat inflexible in the manner in which they select problems to address, and in the manner in which they apply rules to solve them. The rules may themselves be wrong. The synchronous processing scheme can lead to overloading, which causes additional problems. Timing values used in ‘check_’ beliefs also require examination, as the performance assessment and discussion in Callantine (2002b) suggests. Nonetheless, given that the nominal ‘correctly functioning’ agents do perform reasonably well, given the general difficulty of the air traffic control problem, they provide a basis for assessing the performance of the error-generating agents, as presented in the next section.

Performance Assessment

A performance assessment compared the operation of the nominal CATS-based air traffic controller agents to agents with the error mechanisms. Error-generating agents controlled traffic in each of the same three traffic sectors (ADM, SPS, and UKW) used to test the nominal agents in Callantine (2002b). Likewise, the agents controlled traffic for each of the same nine scenarios used in Callantine (2002b). The error-generating agents output descriptive information each time an error mechanism is triggered, so that the effects of the error mechanisms can be examined. This section of the report details the results of this preliminary performance assessment.

The results shown are comparisons between the ‘full control’ condition used in Callantine (2002b), in which the nominal agents issue heading/route, speed, and altitude conditions to control traffic, and the ‘error’ condition in which the agents operate subject to possible errors. Each scenario was run but once in the error condition, so the preliminary assessment presented here only gives a flavor for the sorts of behavior that the agents that err can produce. Numerous Monte Carlo runs would be required to completely characterize agent performance, as well as for the end application of assessing system safety in the face of errors. This research is limited by the lack of a fast-time air traffic simulation with which to conduct such testing. Nonetheless, this section presents results that establish the viability of the error mechanisms implemented in the agents for producing some realistic errors (cf. Durso, et al., 1998).

A second caveat regarding these results, mentioned above, is that the agents in the ‘full control’ condition also do not exhibit perfect performance. The CATS model that structures the overall air traffic control task has not been fully validated, nor have the skills and control rules upon which agent performance depends so heavily. Callantine (2002b) also notes potential problems with the synchronous processing scheme. However, because the agents in the full control condition were directly modified by implementing the above error mechanisms, they provide a convenient baseline against which to assess the performance of the agents that err. The performance assessment section of

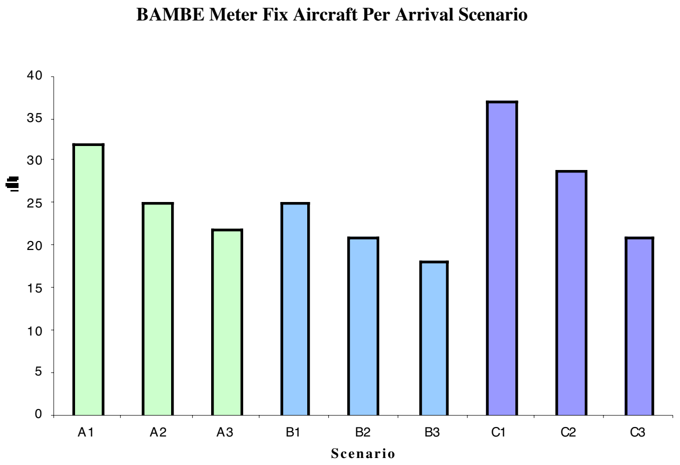

Figure 3. Number of aircraft slated to cross the BAMBE in each traffic scenario.

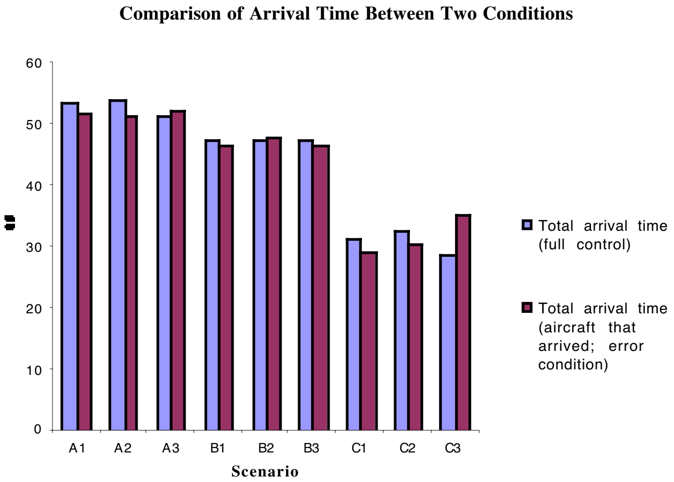

Figure 4. Total time for all aircraft to cross the meter fix by scenario and condition.

Callantine (2002b) presents a comparison of nominal agent performance against a control condition in which agents only issue descent clearances to the required sector exit altitude.

Figure 3 characterizes the nine arrival scenarios (which do not include departures and overflights) in terms of the number of aircraft slated to cross the primary meter fix of interest (‘BAMBE’). Figure 4 graphs the total aircraft arrival time in each scenario for the two conditions. While many of the results seem to indicate that the error-generating agents controlled traffic in such a way that total arrival time was reduced from that in the full control condition, this is tempered by the ‘loss’ of aircraft in the error condition. As Figure 5 shows, the error-generating agents typically failed to make at least a few aircraft successfully cross the meter fix. The agents may have erred in such a way that the aircraft left the airspace improperly, or simply failed to make a last turn to cross the meter fix. Figure 6 shows an example of several sector-boundary violations committed by an erring agent. Such errors are serious if committed by an actual air traffic controller. (Appendix A

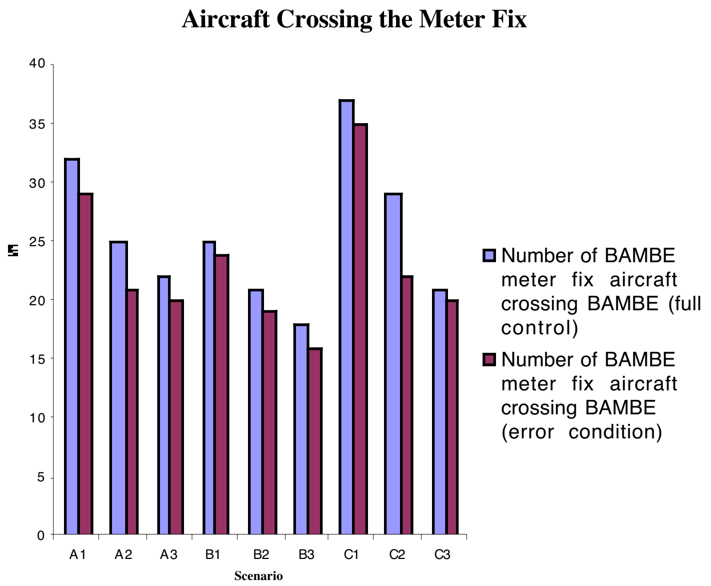

Figure 5. Actual number of aircraft that crossed the meter fix under each condition.

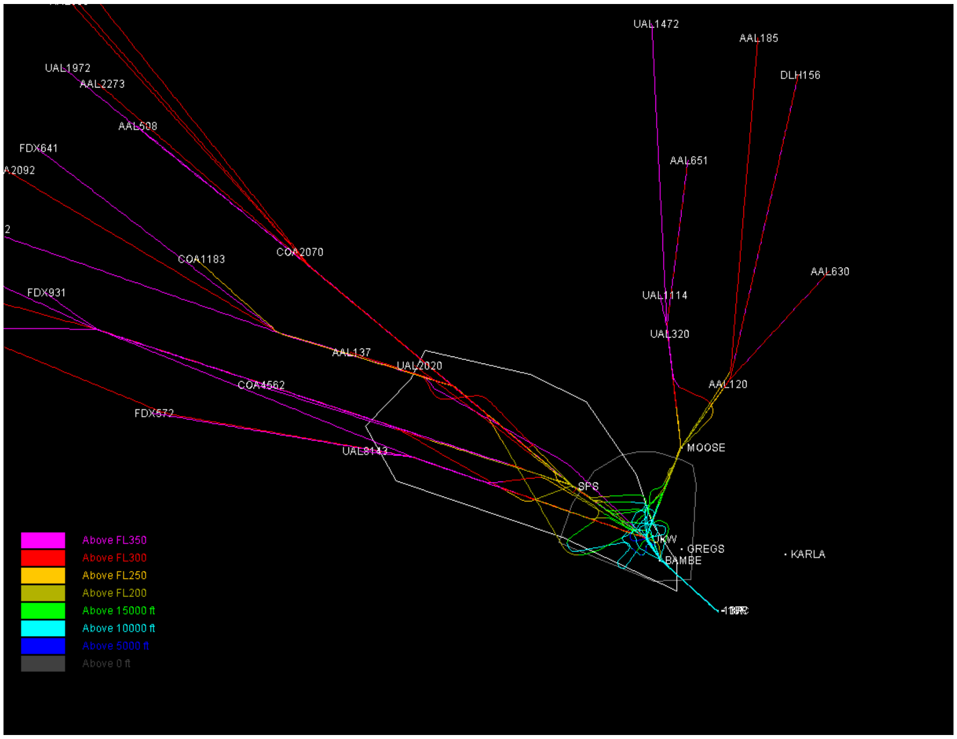

Figure 6. Example traffic trace from scenario A2, showing airspace violations.

provides traces of the traffic flows for all pairs of test scenarios.)

The effect of controlling fewer aircraft as they approach the meter fix figures prominently in interpreting subsequent results. For example, Figure 7 depicts the number of separation violations recorded in each scenario under the full control and error conditions. Only three scenarios registered more separation violations in the error condition than in the full control

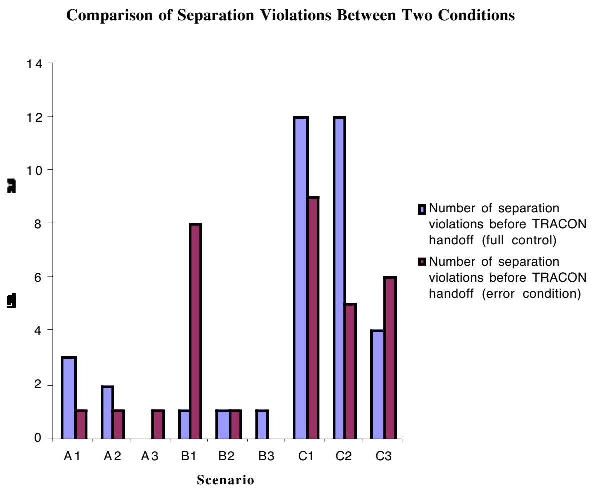

Figure 7. Number of separation violations in each conditions.

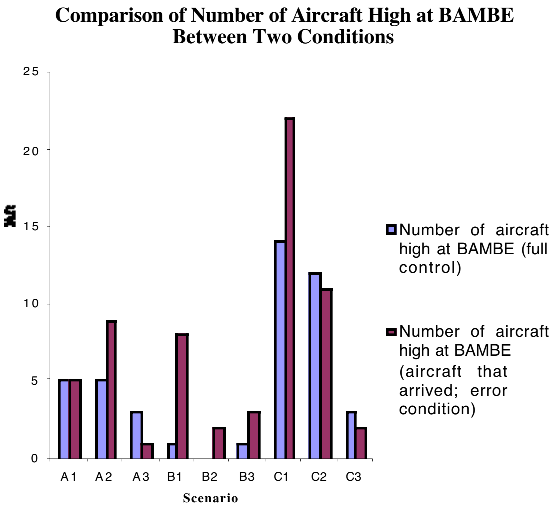

Figure 8. Number of aircraft crossing BAMBE at an improperly high altitude in each condition.

condition. However, depending on where an aircraft is located within the traffic flow, omitting it may have a considerable effect on the difficulty of the merge problem during which most separation violations occur. Airspace violations of the sort shown in Figure 6—while they do not necessarily prevent aircraft from eventually crossing the meter fix—also have the effect of removing the aircraft from the congested region of airspace where aircraft with typical routes are likely to be merging. Given the probabilistic nature of the error mechanisms, these effects are examples of ones in which numerous repeated trials are required to fully assess them.

Figure 8 shows the number of aircraft 'high' at the meter fix across the two conditions. This measure also indicates that the error-generating agents are not consistently worse than the nominal agents at ensuring the aircraft cross the meter fix at the proper altitude. The nominal agents performed better in only five of the nine scenarios, against the single trial for the erring agents. Callantine (2002b) discusses the 'high at the meter fix' results, suggesting that overloading due to

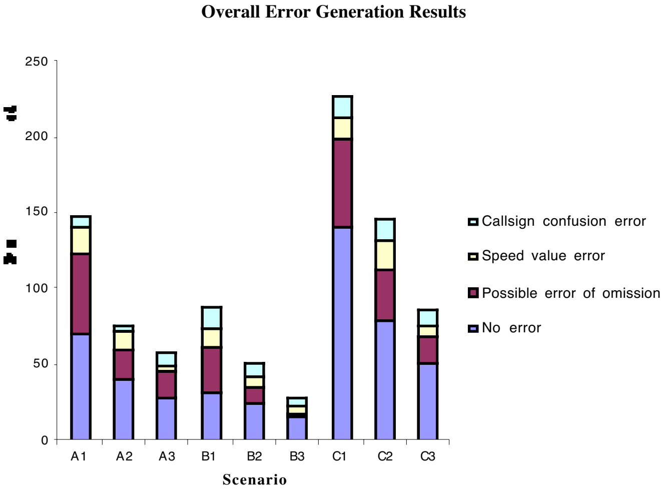

Figure 9. Overall breakdown of outcomes for each triggered error mechanism for all three error-generating agents.

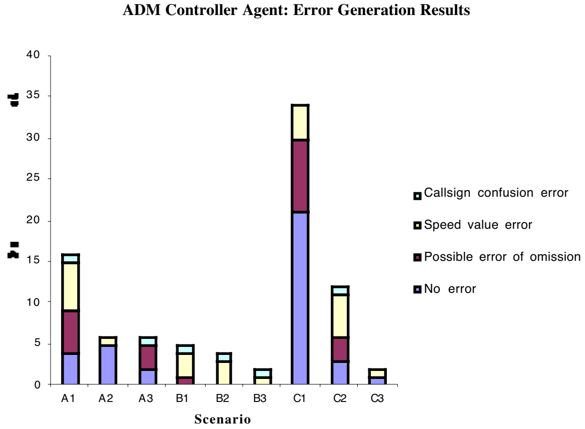

Figure 10. Outcomes for triggered error mechanisms for the ADM agent.

excessive conflicts at the merge point and an inflexible prioritization scheme that prevents agents from descending aircraft play a role in these problems. Again, simply having less aircraft to deal with, after having lost a few, probably helps the erring agents; with less conflicts to deal with, the agents are placed in a position to be able to issue the required descent clearances.

The remaining results concern the performance of the error mechanisms. A detailed analysis of the output data generated by the error mechanisms yields a number of insights about the effectiveness of the mechanisms for yielding errors, as well as evidence to support the notion that the error-generating agents are actually error-tolerant in some respects. The results show the outcome each time an agent successfully triggered an error mechanism.

Figure 9, for example, graphs the outcomes for each scenario for all three agents (i.e., ADM, UKW, and SPS). The total number of triggered error mechanisms ranges from twenty-eight in the lightest traffic scenario (B3) to 228 in the heaviest (C1). In many of these cases, the error mechanism has no effect, because the error is ‘caught’ by an error-tolerant feature of the processing scheme, or because it only affected a belief about a low-priority control problem. As an

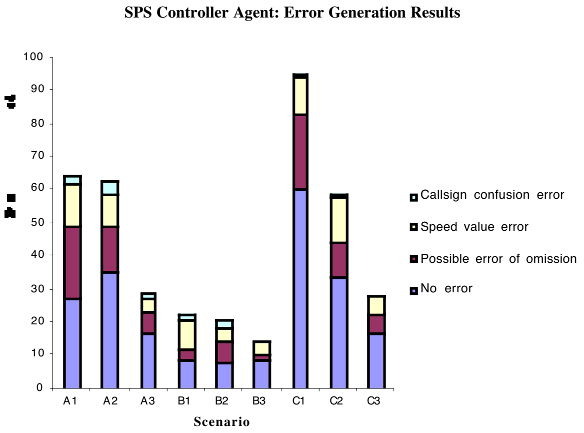

Figure 11. Outcomes for triggered error mechanisms for the SPS agent.

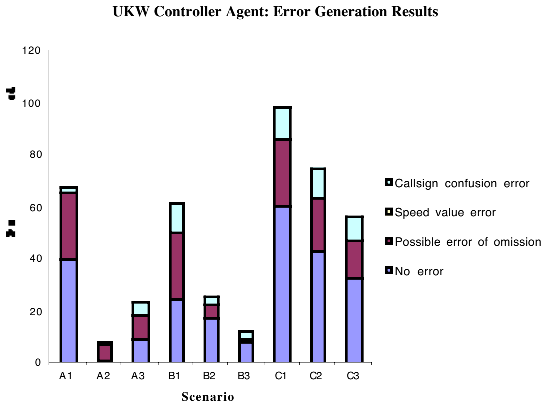

Figure 12. Outcomes for triggered error mechanisms for the UKW agent.

example, consider a case where the agents alters a belief about a ‘within_flow_spacing’ problem by removing the last set of aircraft found to qualify as such. This has no effect if the agent is also dealing with numerous conflicts or other sets of improperly spaced aircraft listed earlier in the ‘within_flow_spacing’ belief. Before the agent misses the set of aircraft, it will have re-established the belief by re-assessing the displayed traffic, and will have again placed the aircraft set in the belief. Another case when ‘no error’ occurs is when multiple errors cancel each other out. An example concerns the ‘misreading’ displayed information or improper recall of speed range performance information. If the reference aircraft’s speed was misread .02 Mach low, but the speed performance range of the aircraft to clear was recalled .02 Mach high, the speed clearance control rule yields a clearance that was exactly what it would have been if the agent made no errors.

The next category of outcome for a triggered error is ‘possible error of omission.’ This outcome signifies, first, the possibility that a set of aircraft in a belief was ‘forgotten’ that the agent would likely have used as the basis for some control action. Second, an agent may have confused a callsign in a situation where the error on the incorrect aircraft was caught by an error-tolerance mechanism. For example, if an agent confused a callsign for a ‘descent_aircraft,’ and the ‘new’ aircraft’s constraints indicate that the aircraft is already descending to the proposed clearance altitude, the agent will do nothing. Meanwhile, the correct aircraft to descend receives no descent clearance—an error of omission. Unfortunately, not all context information necessary to accurately flag errors of omission is available (hence the designation, 'possible error of omission'). The indicator for this classification is defined as an error (i.e., forgetting or swapping a callsign) in the first set of aircraft in a belief. It is this first set of aircraft that the agent would have likely addressed shortly if the error did not intervene.

Finally, two categories of outcomes signify errors of commission with verified effects. They are designated 'speed value error' and 'callsign confusion error.' Speed value errors occur when the agent issues a speed clearance that is modified by the 'misread or incorrectly recall' error mechanism and the results are not subject to the cancellation effect noted above. Callsign confusion errors do not refer to the 'callsign confusion' error mechanism—they result when the agent confuses a callsign when manipulating a belief or confuses the front and back aircraft in a control rule.

Interestingly, the fourth error mechanism (confusing the clearance type and contents) was never manifested. An examination of the error mechanism indicates that this is due to characteristics of the test scenarios and agent performance. First, the SPS and ADM agents primarily use Mach numbers when issuing speed clearances; however, the error mechanism does not confuse a Mach number with another clearance type (due to the presence of the decimal that always appears for subsonic flight). The low-altitude controller (UKW) is so overloaded by merging aircraft that it hardly has time to issue speeds (although it should). The error mechanism only triggers with probability .05 in these cases, and with only nine total trials, the effect was never observed. This indicates that, at least, heading values should have been confused with speeds. However, the nature of the test airspace (arrivals from the west and north) makes intended heading values that can be confused with speeds extremely unlikely. Indeed, heading values from 240 to 300 that might be confused with airspeeds are in the opposite direction of the general flow of arrival traffic in the test scenarios, so the agents probably never considered such a clearance, and this error mechanism never had a chance to create an error. Thus, further research is required to assess its performance.

Returning now to Figure 9, which depicts the overall outcomes from triggered error mechanisms, the analysis shows that the most typical outcome is 'no error.' The second most prevalent outcome is 'possible error of omission.' Speed value and callsign confusion errors, taken together, are third. The number of overall mechanisms triggered, and the proportion of relevant effects indicates that, as implemented, a probability value of .05 is too high. This is because experienced air traffic controllers make relatively few errors. However, for research purposes, the .05 value may be acceptable for reducing the total number of trials necessary to assess the robustness and safety of a given ATC concept.

The remaining graphs (Figures 10, 11, and 12) show the contribution of the individual agents to the overall results. Figures 10 and 11 show, for the high-altitude sectors (ADM and SPS), speed value errors occur more often than callsign confusion errors, while in the UKW low-altitude sector (Figure 12), callsign confusion errors dominate. This effect is due to the lack of speed clearances issued by UKW, whose primary job is to issue heading vectors to solve the merge problem. The same 'conflict overload' condition observed previously for the UKW agent (Callantine, 2002b) means the agent never has time to address spacing problems.

In summary, the results indicate that the preliminary set of error mechanisms this research identified work to produce some realistic errors within the CATS agent framework. There are no doubt others; further research is required to implement and assess additional alternative error mechanisms. Some features of the framework, including those that nullify the effects of error mechanisms, deserve closer examination. Furthermore, a more detailed analysis is required to ensure that the number and relative proportions of errors are suitable for assessing safety risks of ATC concepts, while keeping the overall number of errors produced in given trial relatively low. It may be, for example, that actual air traffic controllers commit errors of omission with far greater regularity than other types of errors or that they are virtually always corrected in time. Another issue is whether errors of omission actually reduce the tendency to 'over-control' traffic occasionally observed with actual air traffic controllers, as well as the nominal agents. If so, the errors of omission noted here may, in some cases, have the effect of actually improving the performance of the agents by removing the tendency to over-control traffic.

Conclusion

This report builds upon the research presented in Callantine (2002b) to offer a method by which CATS-based air traffic controller agents can make realistic errors. The research identifies a number of issues with the approach, both in terms of the agent architecture, processing scheme, and knowledge representations, and for the larger question about using such an approach to analyze system safety. Additional evaluation and validation is needed, but the research has produced enough evidence to enumerate several avenues for further research. First, for the agents themselves, a better understanding of the effects of error detection and resolution is needed. As noted above, many errors are thwarted before they are manifested in an incorrect clearance or lack thereof, and errors of different classes are created in proportions that may or may not correspond to real-world effects. Second, additional research is needed to understand interactions between different errors and develop methods for tracing their impact on the system. This area of research addresses the issue of how errors 'chain' to eventually compromise safety (Reason, 1990). For both of these issues, a fast-time air traffic simulation would be helpful for obtaining more results. In conclusion, this preliminary research indicates that a CATS-based framework for agents that err is viable, and that further research on agents that err for safety assessment is warranted.

References

Callantine, T. (2001). Agents for analysis and design of complex systems. *Proceedings of the 2001 International Conference on Systems, Man, and Cybernetics*, October, 567-573.

Callantine, T. (2002a). A representation of air traffic control clearance constraints for intelligent agents. In A. El Kamel, K. Mellouli, and P. Bourne (Eds.), *Proceedings of the 2002 IEEE International Conference on Systems, Man, and Cybernetics*, #WA1C2, CD-ROM.

Callantine, T. (2002b). *CATS-based air traffic controller agents*. NASA Contractor Report 2002-211856, Moffett Field, CA: NASA Ames Research Center.

Degani, A. and Heymann, M. (2000). *Some formal aspects of human automation interaction*. NASA Technical Memorandum 209600, Moffett Field, CA: NASA Ames Research Center.

Durso, F., Truitt, T., Hackworth, C., Crutchfield, J., and Manning, C. (1998). En route operational errors and situation awareness. *International Journal of Aviation Psychology, 8*(2), 177-194.

Endsley, M. (1999). Situation awareness and human error: Designing to support human performance. *Proceedings of the High Consequence Systems Surety Conference*, Albuquerque, NM.

Fields, R., Harrison, M., and Wright, P. (1997). THEA: Human error analysis for requirements definition. Technical Report 2941997, York, UK: University of York Computer Science Department.

Funk, K. (1991). Cockpit task management: Preliminary definitions, normative theory, error taxonomy, and design recommendations. *International Journal of Aviation Psychology, 1*(4), 271-285.

Hollnagel, E. (1991). The phenotype of erroneous actions: Implications for HCI design. In G. R. S. Weir and J. L. Alty (Eds.), *Human-Computer Interaction and Complex Systems*. London: Academic Press, 73-121.

Jones, P., and Mitchell, C. (1987). Operator modeling: Conceptual and methodological distinctions. *Proceedings of the Human Factors Society 31st Annual Meeting*, 31-35.

Norman, D. (1981). Categorization of action slips. *Psychological Review, 88*(1), 1-15.

Norman, D. (1988). *The psychology of everyday things*. New York: Basic Books.

O'Hare, D., Wiggins, M., Batt, R., and Morrison, D. (1994). Cognitive failure analysis for aircraft accident investigation. *Ergonomics, 37*(11), 1855-1869.

Pocock, S., Harrison, M., Wright, P., and Johnson, P. (2001). THEA: A technique for human error assessment early in design. In M. Hirose (Ed.), *Human-Computer Interaction: INTERACT'01*, Amsterdam: IOS Press, 247-254.

Polson, P., Lewis, C., Rieman, J., and Wharton, C. (1992). Cognitive walkthroughs: A method for theory-based evaluation of interfaces. *International Journal of Man-Machine Studies, 36*, 733-741.

Rasmussen, J. (1982). Human errors: A taxonomy for describing human malfunction in industrial installations. *Journal of Occupational Accidents, 4*, 311-333.

Reason, J. (1990). *Human error*. New York: Cambridge Press.

Shappell, S. and Wiegmann, D. (1997). A human error approach to accident investigation: The taxonomy of unsafe operations. *International Journal of Aviation Psychology, 7*(4), 269-291.

Wickens, C. and Flach, J. (1988). Information processing. In E. L. Wiener and D. C. Nagel (Eds.), *Human Factors in Aviation*, San Diego, CA: Academic Press, 111-155.

Wiegmann, D. and Shappell, S. (1997). Human factors analysis of post-accident data: Applying theoretical taxonomies of human error. *International Journal of Aviation Psychology, 7*(1), 67-81.

Wiegmann, D. and Shappell, S. (2001). Human error perspectives in aviation. *International Journal of Aviation Psychology, 11*(4), 341-357.

Appendix A

This appendix shows traffic flows for each of the scenarios in each of the two conditions. The traffic traces are shown in pairs, with the baseline agents on the top, and the erring agents on the bottom.

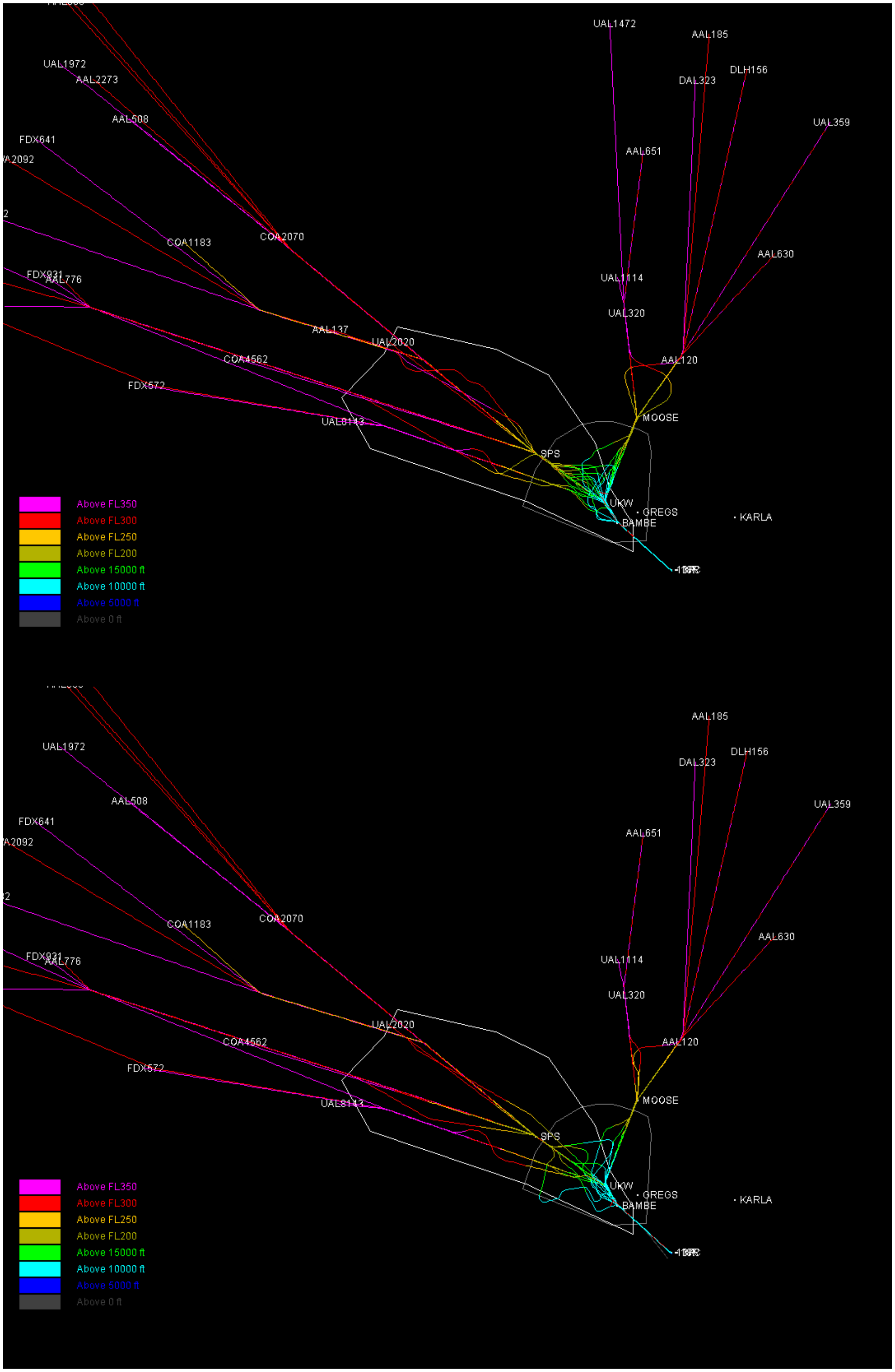

Figure B-1. Flows for scenario A-1.

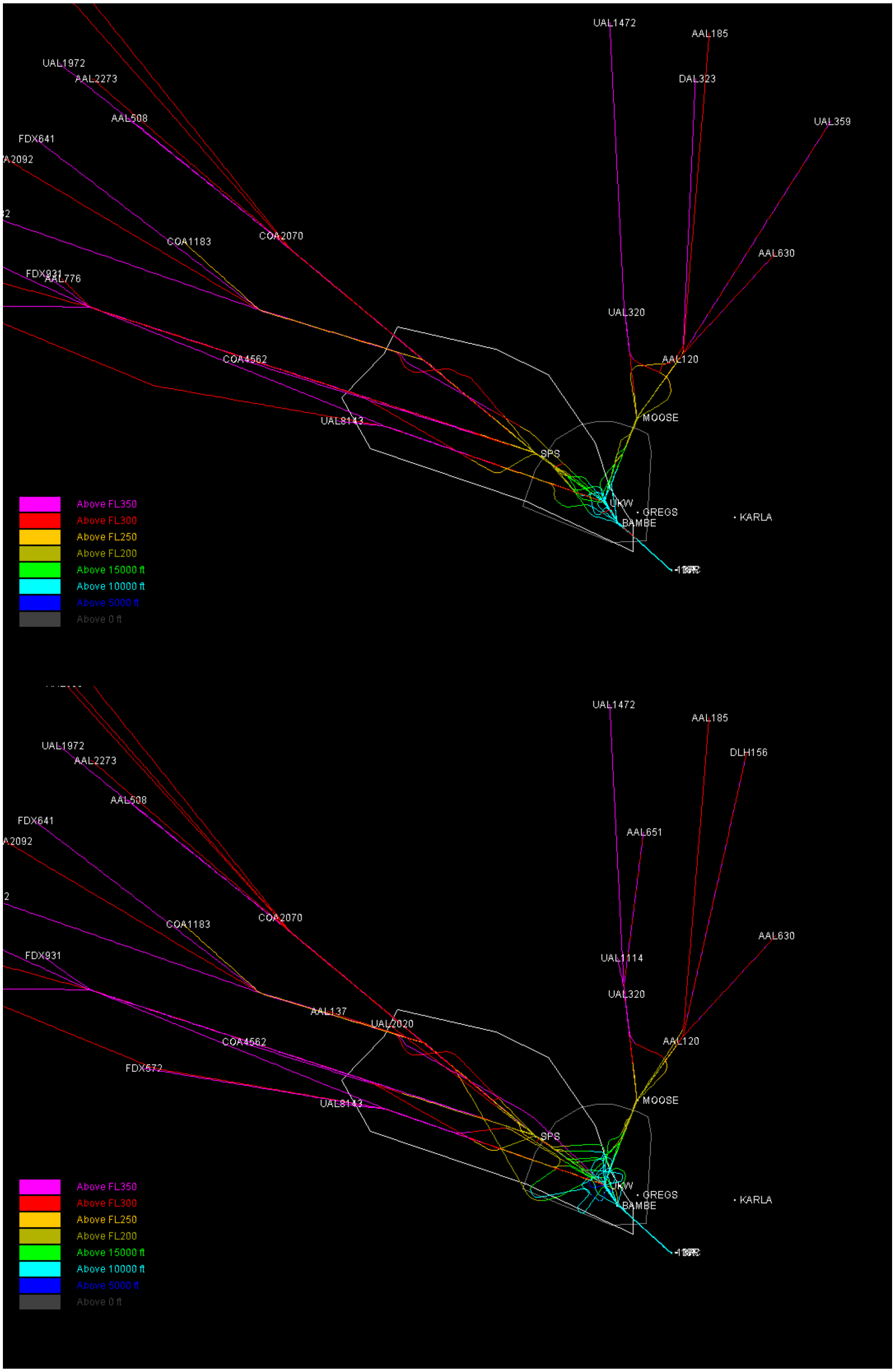

Figure B-2. Flows for scenario A-2.

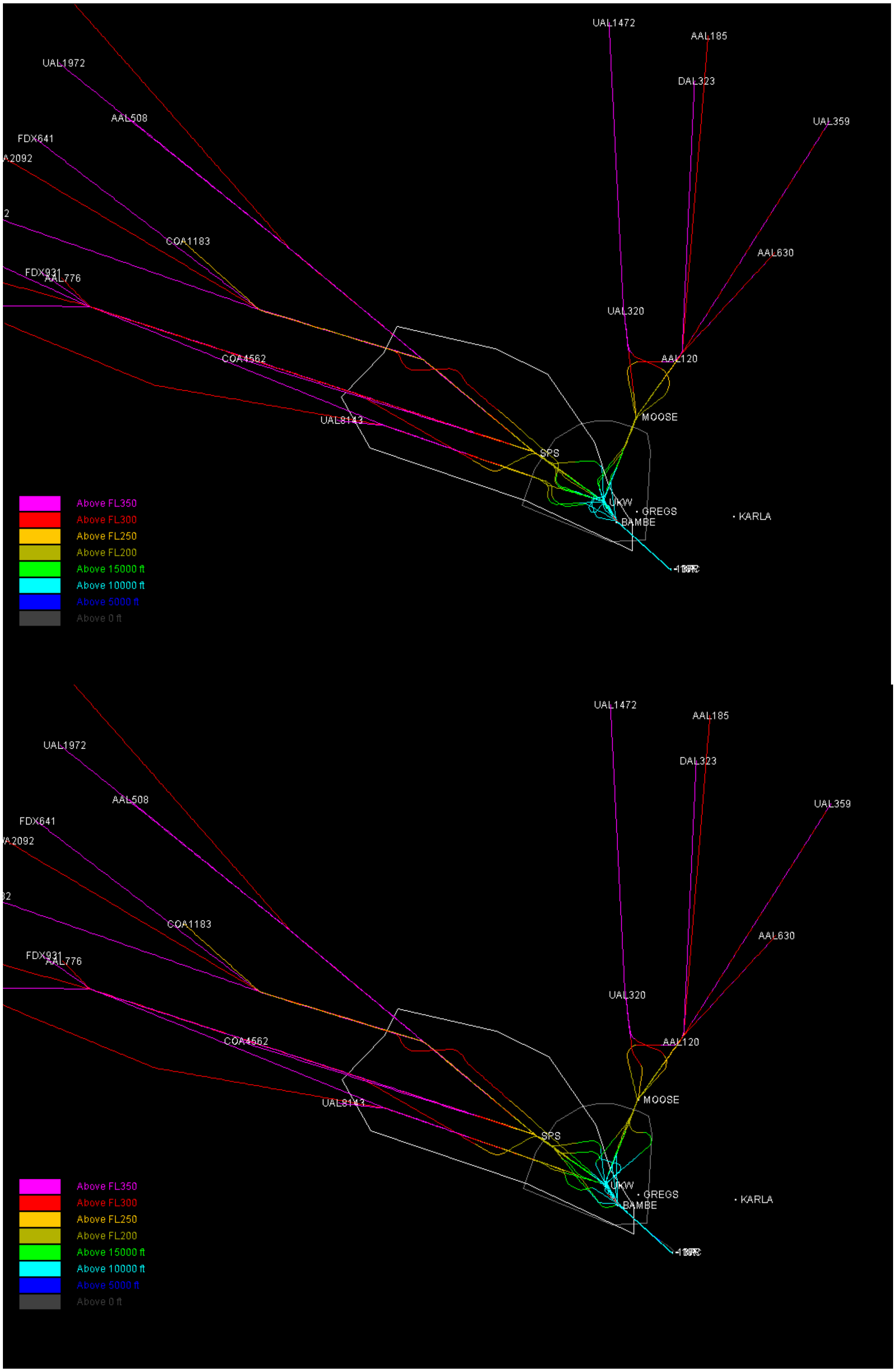

Figure B-1. Flows for scenario A-3.

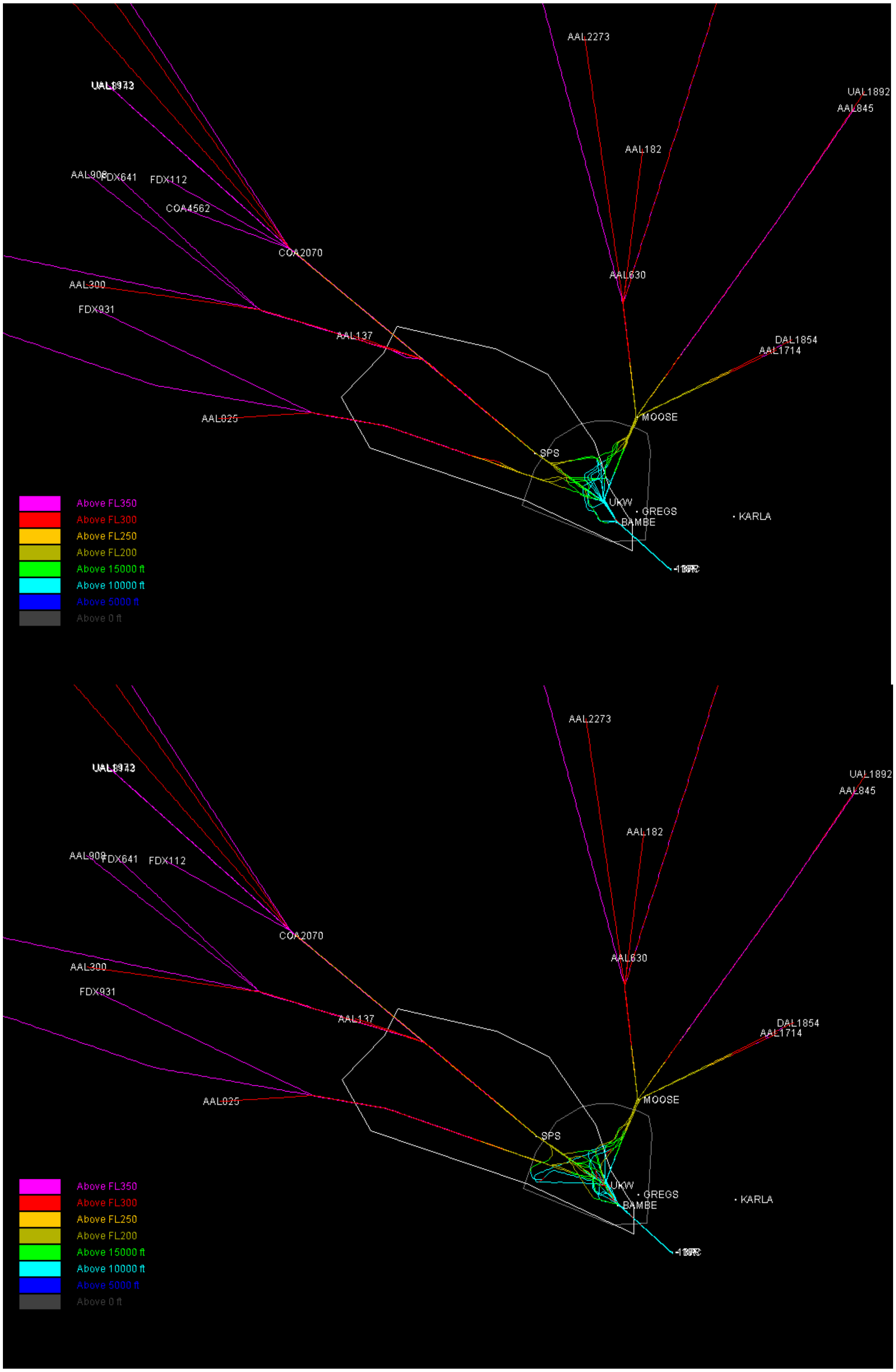

Figure B-4. Flows for scenario B-1.

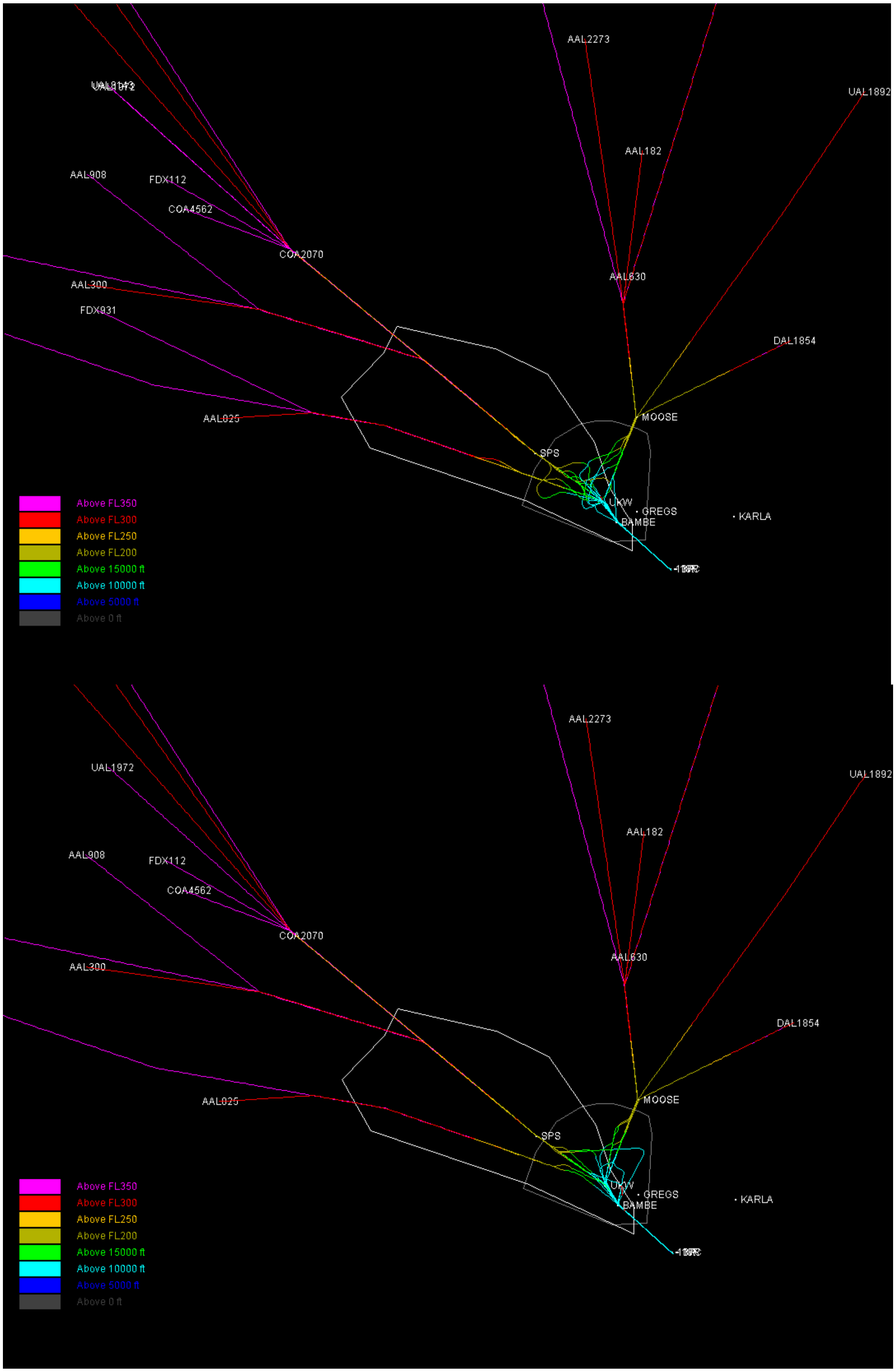

Figure B-5. Flows for scenario B-2.

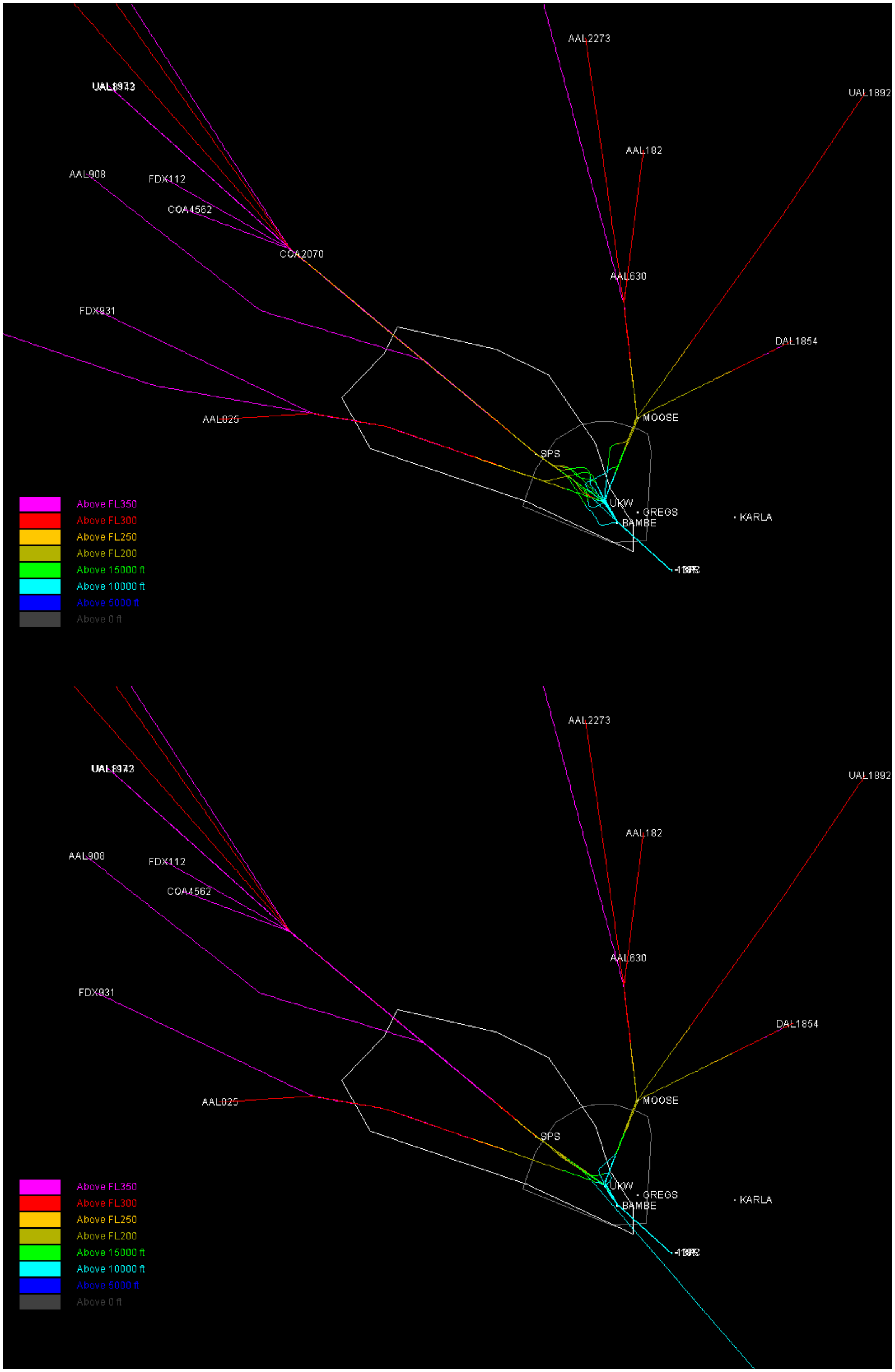

Figure B-6. Flows for scenario B-3.

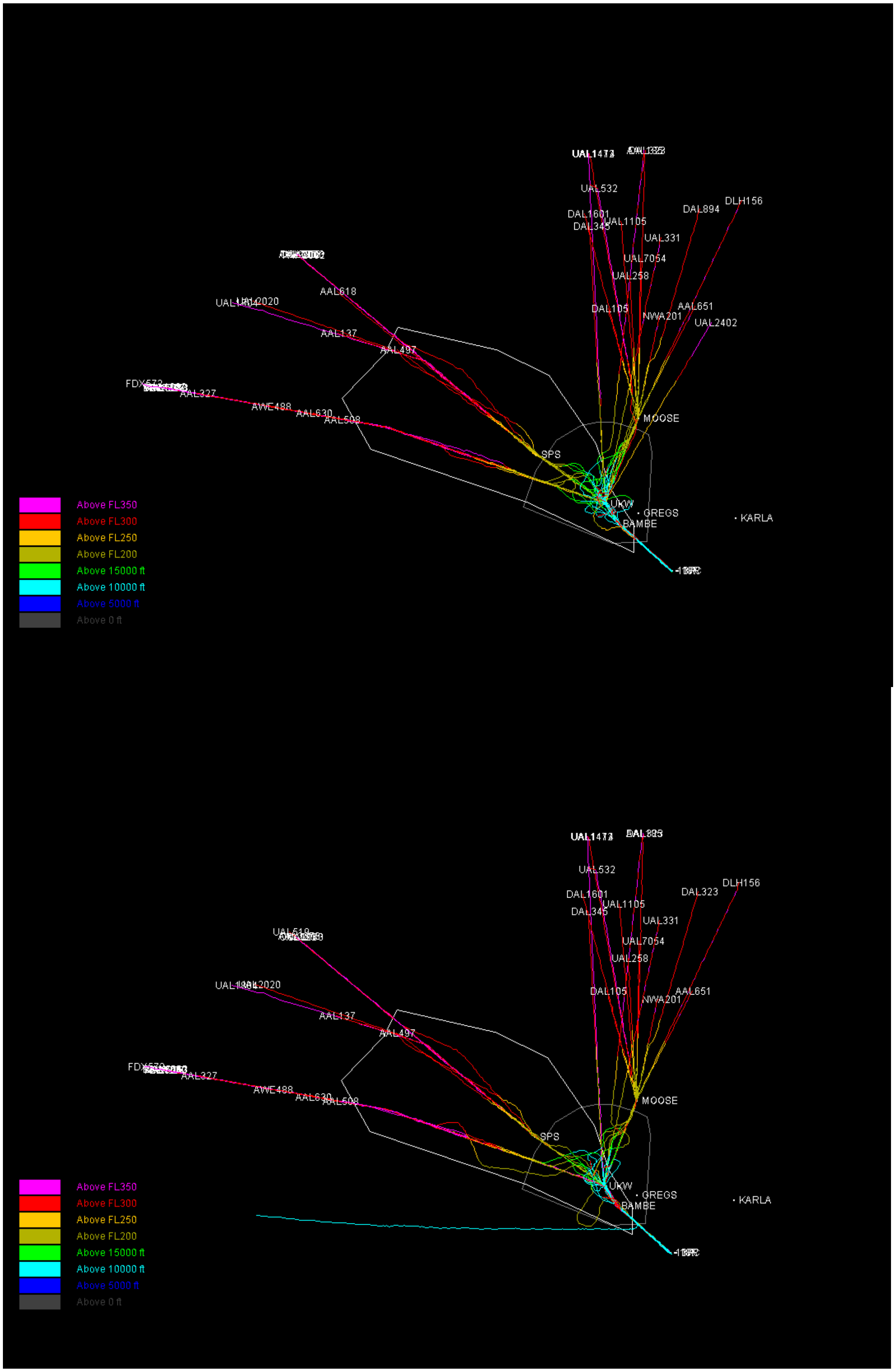

Figure B-7. Flows for scenario C-1.

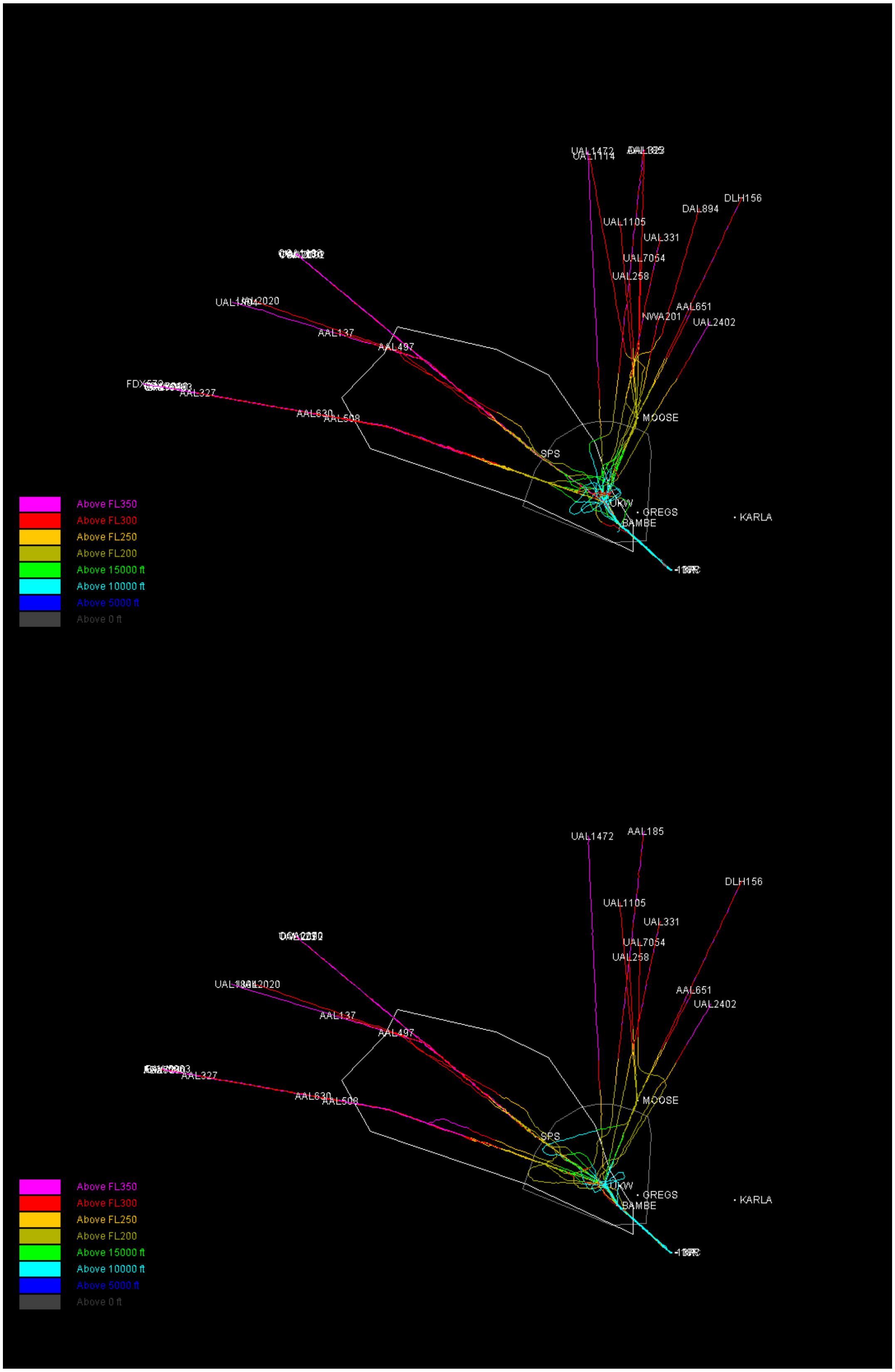

Figure B-8. Flows for scenario C-2.

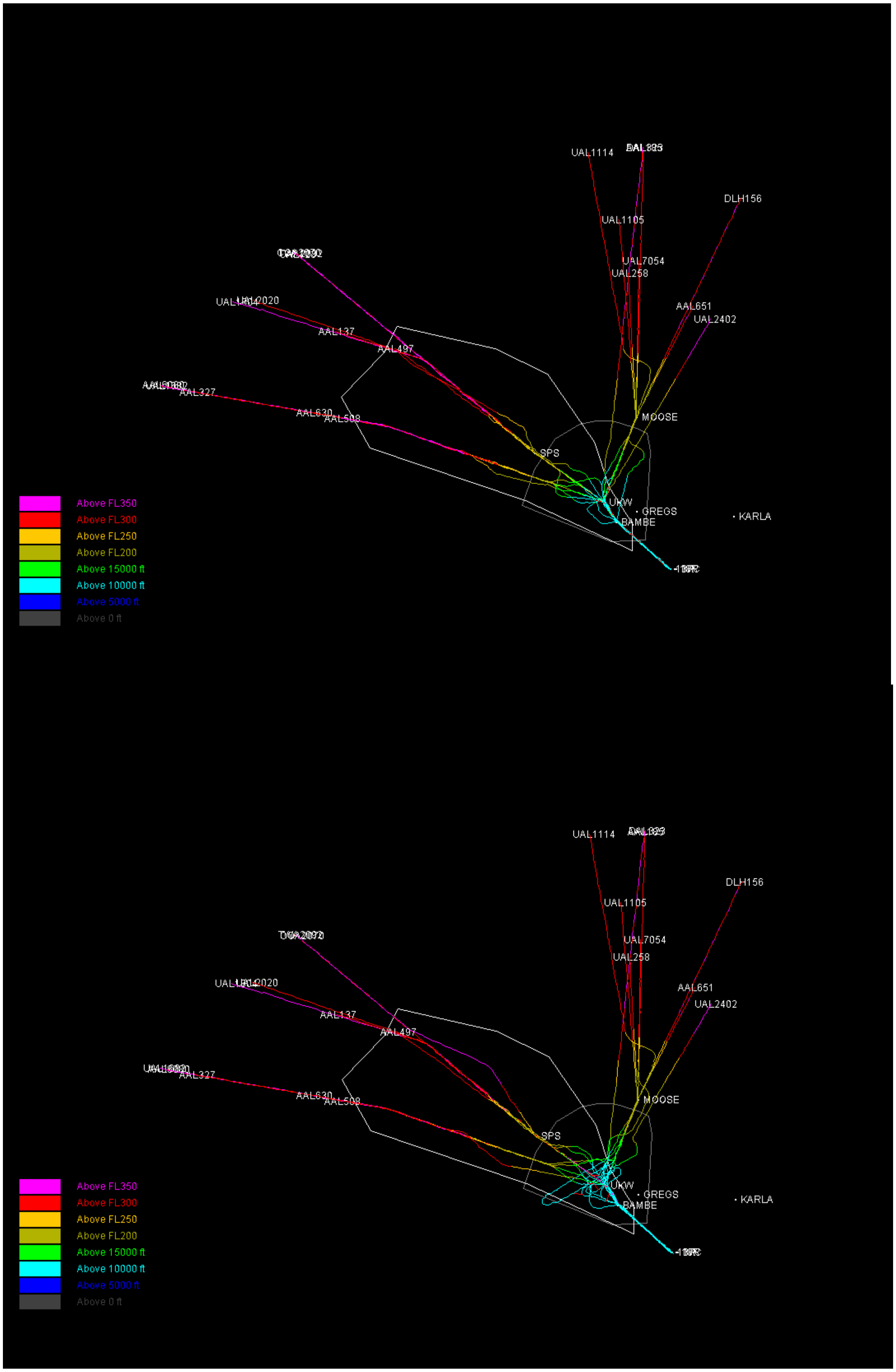

Figure B-9. Flows for scenario C-3.